IDA PAPER P-2895

SOFTWARE CAPABILITY EVALUATIONS INCORPORATING TRUSTED SOFTWARE METHODOLOGY

Dennis W. Fife, *Task Leader*

Judy Popelas
Beth Springsteen

October 1994

Prepared for
Ballistic Missile Defense Organization

INSTITUTE FOR DEFENSE ANALYSES
1801 N. Beauregard Street, Alexandria, Virginia 22311-1772

DEFINITIONS

IDA publishes the following documents to report the results of its work.

Reports

Reports are the most authoritative and most carefully considered products IDA publishes. They normally embody results of major projects which (a) have a direct bearing on decisions affecting major programs, (b) address issues of significant concern to the Executive Branch, the Congress and/or the public, or (c) address issues that have significant economic implications. IDA Reports are reviewed by outside panels of experts to ensure their high quality and relevance to the problems studied, and they are released by the President of IDA.

Group Reports

Group Reports record the findings and results of IDA established working groups and panels composed of senior individuals addressing major issues which otherwise would be the subject of an IDA Report. IDA Group Reports are reviewed by the senior individuals responsible for the project and others as selected by IDA to ensure their high quality and relevance to the problems studied, and are released by the President of IDA.

Papers

Papers, also authoritative and carefully considered products of IDA, address studies that are narrower in scope than those covered in Reports. IDA Papers are reviewed to ensure that they meet the high standards expected of refereed papers in professional journals or formal Agency reports.

Documents

IDA Documents are used for the convenience of the sponsors or the analysts (a) to record substantive work done in quick reaction studies, (b) to record the proceedings of conferences and meetings, (c) to make available preliminary and tentative results of analyses, (d) to record data developed in the course of an investigation, or (e) to forward information that is essentially unanalyzed and unevaluated. The review of IDA Documents is suited to their content and intended use.

The work reported in this document was conducted under contract DASW01 94 C 0054 for the Department of Defense. The publication of this IDA document does not indicate endorsement by the Department of Defense, nor should the contents be construed as reflecting the official position of that Agency.

IDA PAPER P-2895

SOFTWARE CAPABILITY EVALUATIONS INCORPORATING TRUSTED SOFTWARE METHODOLOGY

Dennis W. Fife, *Task Leader*

Judy Popelas
Beth Springsteen

October 1994

INSTITUTE FOR DEFENSE ANALYSES

Contract DASW01 94 C 0054
Task T-R2-597.2

PREFACE

This paper was prepared by the Institute for Defense Analyses (IDA) under the task order, Ballistic Missile Defense (BMD) Software Technology, and fulfills an objective of the task, to "prepare a final guide for BMD Software Capability Evaluation (SCE) teams based on available BMD experience . . . in applying IDA's FY 93 draft guide incorporating the BMD Trusted Software Development Methodology (TSDM)." The work was sponsored by the Ballistic Missile Defense Organization (BMDO).

The following IDA research staff members were reviewers of this document: Mr. Bill Brykczynski, Dr. Richard J. Ivanetich, Mr. Terry Mayfield, Dr. Reginald N. Meeson, and Dr. D. Robert Worley. Their contributions are gratefully acknowledged.

Table of Contents

List of Figures

List of Tables

SUMMARY

The Ballistic Missile Defense Organization is using both the Ballistic Missile Defense (BMD) Trusted Software Methodology (TSM) and the Software Engineering Institute (SEI) Process Maturity Model (PMM) to identify risks on software-intensive development programs. This paper provides an evaluation approach that supplements the PMM with criteria from the TSM and incorporates it into the Software Capability Evaluation (SCE) process with the least disruption to established practice. The results of the study are intended to be used by team members already trained in both the TSM criteria and the SCE methodology. The results can also serve as a general example for extending the SEI criteria with software evaluation models other than the TSM.

Overview of the Trusted Software Methodology

The BMDO sponsored the development of the TSM to help reduce the potential for inserting malicious and inadvertent flaws into BMD software. The TSM defines 25 Trust Principles that represent safeguards or countermeasures to reduce threats or vulnerabilities to the software. The BMD Software Directive 3405 requires each Program Manager to assess the risks of not complying with each Trust Principle, identify options to mitigate the risk of non-compliance, or accept the program risks associated with non-compliance.

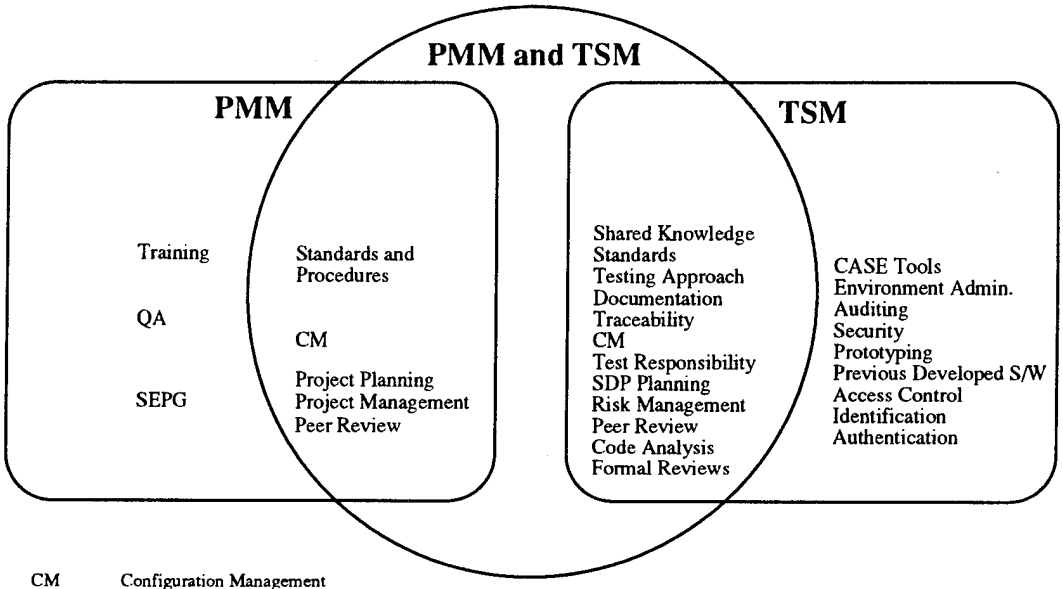
Overview of the Process Maturity Model

The PMM is used to evaluate and rank the maturity of contractor's software development processes. The contractor's process is evaluated by an SCE team of software experts who identify the strengths and weaknesses of a contractor's process relative to the PMM criteria which incorporate good software engineering practices. SCEs involve a three-day site visit at the contractor's facility. The evaluation team conducts interviews and documentation reviews to assess the quality of the contractor's software development process. SCEs have become widely used throughout the Department of Defense as a means of identifying software risks early and for providing input to the Source Selection Evaluation Board. Hence, BMD Software Directive 3405 now requires the Program Elements to per-

form SCEs for all source selections and to use SCEs to help monitor a contractor's process improvement throughout the life of the program. Over eight SCEs have been completed within the BMD program and there are plans to perform many more over the next few years.

Recommended TSM/PMM Model

The new TSM/PMM model was developed to help eliminate duplication between an evaluation based on the TSM and one based on the PMM, and to streamline the criteria for the SCE teams. The TSM and PMM both revolve around a hierarchial model consisting of five levels of trust or process maturity, but the BMD program intends to evaluate a contractor's development risks relative only to level three for both of these models. The Venn diagram in Figure S-1 identifies the areas of overlap and the differences between the two models for levels 1, 2, and 3. As illustrated, many of the PMM key process areas (KPAs) are similar to the TSM Trust Principles. A new model was formed by combining the Trust Principles and the KPAs that are similar and grouping the unique Trust Principles separately. Table S-1 illustrates the new TSM/PMM model which consists primarily of 10 KPAs; 8 of these are from the PMM, and 2 new KPAs were formed from the unique TSM requirements.



CM Configuration Management
CASE Computer-Assisted Software Engineering
QA Quality Assurance
S/W Software
SDP Software Development Plan
SEPG Software Engineering Process Group

Figure S-1. Overlap Between PMM and TSM

Recommended Revisions to the Evaluation Process

With the addition of the TSM criteria to the SCE process, several changes must be made to the activities typically conducted prior to the site visit, during the site visit, and at the close of the site visit. Prior to the site visit, the Requests for Proposal must state that a TSM/PMM evaluation will be performed. During the site visit, the SCE team will request additional documentation and perform several new interviews to account for the additional TSM/PMM criteria. At the end of the site visit, the format of the team's findings will also be adjusted to incorporate the new model criteria. This document contains detailed evaluation criteria for the SCE team and descriptions of the additional activities that must be performed during the SCE to account for the TSM/PMM requirements.

Table S-1. TSM/PMM Model

SEI KPA	Added Trust Principles Consistent with KPAs	New Trust KPAs	Trust Principles Unique to TSM
Software Engineering Process Group	(none)	Secure Development Environment	Access Control (T3) Auditing (T3) CASE tools (T2) Environment Administration (T1) Identification & Authentication (T2)
Training	(none)		
Peer Reviews	Shared Knowledge (T3) Peer Review (T1)		
Standards & Procedures	Testing Approach (T3) Standards (T2) Documentation (T1)		
Quality Assurance	(none)	Project Trust Policy	Distribution (T3) Security (T2) Prototyping (T1) Previous Developed Software (T1)
Configuration Management	Traceability (T2) Configuration Management (T1)		
Project Planning	Test Responsibility (T3) Software Development Plan (SDP) Planning (T1)		
Project Management	Code Analysis (T3) Formal Reviews (T2) Risk Management (T1)		

General Considerations for Extending the PMM

While this document focuses on extending the SEI process maturity model with TSM requirements, other models may be combined with the SEI criteria to meet specific program needs. Examples of other software evaluation models include the Software Development Capability/Capacity Review (SDC/CR), Software Productivity Research (SPR), ISO 9000-3, and Trillium [AFSC 1991; SPR 1991; ISO 1991; Bell 1992]. Additional areas of investigation included in SDC/CR but not found in the SEI KPAs are systems engineering and development tools. SPR has additional areas of coverage such as the physical environment, experience of staff, and development methodologies. Trillium includes systems engineering, reliability engineering, and customer support. ISO 9000-3 has requirements for servicing and purchaser-supplied products.

If an evaluation team is going to perform a dual evaluation, it should first identify requirements that are common between the models and those that are unique to each model. This will help to expedite the on-site evaluation process, eliminate duplication of effort during the interviews and documentation reviews, and maintain consistency between evaluation results. Even though most of the pertinent models have various levels of criteria for evaluating a contractor, specific criteria may be classified at different levels for different models. Since there is no such thing as a "universal level 3," the program manager should select the desired level for each model individually with the assistance of the evaluation team.

The SCE process will also be affected when combining SEI's criteria with those of other models. The evaluation team must make the proper changes to the Request for Proposal, questionnaires, documentation requests, and interview schedules. It is recommended that the summary of evaluation results identify a contractor's level of success for each of the models it is evaluated against. It is important for program management to recognize, for example, that a contractor may have failed to satisfy ISO criteria but at the same time satisfied SEI and TSM level 2 requirements.

1. INTRODUCTION

1.1 Purpose

This study defines a method for evaluating a contractor's ability to comply with both the Software Engineering Institute (SEI) Process Maturity Model (PMM) and the Trusted Software Methodology (TSM) of the Ballistic Missile Defense (BMD). During the last three years, the BMD program has been assessing contractor's software development practices using the SEI process for conducting Software Capability Evaluations (SCEs). Due to the success of the SCE program and its commonality with the TSM, the Institute for Defense Analyses (IDA) was tasked to define an approach for combining the TSM criteria with the PMM criteria and incorporating it into the SCE process with the least disruption to established practice.

1.2 Scope

This paper is for use within the BMD program by teams already trained in the SCE process and the TSM. The paper supplements existing SEI and Ballistic Missile Defense Organization (BMDO) training material. The paper does not explain in detail the SCE process but concentrates on the additional activities and criteria that must be considered when evaluating the TSM requirements along with the PMM.

The term PMM is used to characterize the original SEI maturity model which existed prior to the advent of the SEI Capability Maturity Model (CMM). The PMM is the subject of this paper; the CMM had not been included in the SCE process or the SEI training material at the time this study was written.

1.3 Approach

The following steps were taken in preparation for the analyses in this paper.

a. Trained IDA personnel on SCE process.

   Seven IDA personnel attended the SEI three-day training course for conducting an SCE. The course introduces the PMM and teaches the evaluators techniques for planning the site visit, conducting interviews, and performing documenta-

tion reviews. SEI uses case studies and mock evaluations to provide initial hands-on experience for the trainees.

b. Participated in conducting SCEs.

IDA participated in eight SCEs that were performed on the competing contractors for Brilliant Eyes, Brilliant Pebbles, and the National Test Facility programs. The IDA members of the SCE team helped organize the evaluation teams, coordinated with the contractors and program offices, and acted as technical advisors providing expertise in the area of software development.

c. Trained IDA personnel on TSM criteria.

Five IDA personnel attended the Martin Marietta TSM course which introduced the TSM methodology and evaluation criteria. The case studies that identified potential threat scenarios were used to help the trainees understand the value of the Trust Principles.

d. Developed recommended procedures and techniques.

The SEI and Martin Marietta training material and the experiences in conducting SCEs served as the basis for this document. A model was developed to identify the commonality and uniqueness between the SEI key process areas (KPAs) and the Trust Principles. Key criteria and evaluation practices were then identified for each Trust Principle and summarized for the evaluation team.

e. Planned future use and evolution of TSM/PMM technique.

Plans have been made to beta test the initial TSM/PMM methodology. The results of the beta test and future revisions to the SCE process will be incorporated into the methodology as they evolve. The new SEI maturity model, CMM, has several more KPAs and further refines the evaluation criteria at higher maturity levels. As of this date, SEI has yet to incorporate the CMM into the SCE training course. Once that is accomplished, the TSM/PMM evaluation approach will be updated to include the revisions to the SCE process based on the new CMM.

1.4 Organization

The paper is organized into six sections. Section 1 introduces the paper. Section 2 presents brief overviews of both the BMD TSM and the SEI PMM. Section 3 describes the TSM/PMM model which identifies a means of combining the two sets of evaluation crite-

ria, i.e., the TSM and the PMM. Section 4 describes additional activities that will be added to the SCE process in order to assess the Trusted Software criteria. Section 5 contains information on how the results would be presented. The appendices provide plans and worksheets the evaluation teams will use to conduct the SCE. References and an acronym list are also provided.

2. BACKGROUND

This section of the paper provides a brief overview of both the TSM and the PMM. For additional details, refer to [BMD 1993; Fife 1992; SEI 1992a].

2.1 Overview of the TSM

In 1991, BMDO established the TSM to reduce the potential for inserting flaws into the BMD software. TSM was developed to address both malicious and inadvertent types of flaws. The TSM defines 25 Trust Principles in a hierarchial model consisting of 6 levels, ranging from 0 to 5. Each Trust Principle represents a safeguard or countermeasure that can be implemented to reduce threats or vulnerabilities to the software. At the lowest Trust Levels, the Trust Principles help to guard against inadvertent errors while the Principles introduced at the higher levels help to guard against malicious errors. The TSM Principles were developed from existing software engineering and security requirements, including DOD 5200.28-STD (the Orange Book), DOD-STD-2167A, DOD Directive 5200.28, BMD Software Standards, and software development practices as defined in the SEI PMM. Following is a general characterization of each of the Trust Levels as defined in [GPALS 1992]. Table 1 contains a list of the Trust Principles at their associated levels

- Level T0 is the lowest Trust Level and is associated with software that fails to meet the requirements for the next higher class.
- Level T1 includes administrative policies and procedures, reviews, and documentation. These requirements provide a minimal degree of enhancement to a given development approach with minimal impact.
- Level T2 offers a set of enhancements to an existing DOD-STD-2167A approach intended to reduce the occurrence of inadvertent errors but not intended to counter malicious attacks. The enhancements include requirements for reuse of software, use of automated tools, documented traceability, and coding standards.

- Level T3 introduces minimal requirements necessary to begin preventing malicious errors and includes approaches that have been promoted by the software engineering and quality communities. The criteria enhance management and administrative policy, risk mitigation, prototyping, and testing approaches.
- Level T4 provides additional support in preventing malicious attacks. It includes provisions to ensure a separation of duty and basic automated mechanisms for access control, auditing, and environment integrity checking.
- Level T5 requires the most rigorous software development approach that characterizes current research in software engineering, reliability, security, and quality.

To determine if a Trust Principle is met by a particular software development approach, an evaluator must examine the development process with respect to the compliance criteria that are defined for each of the Trust Principles. For each Trust Principle there are compliance criteria to assure consistent interpretation of trust requirements by the software development and software evaluation organizations. The compliance criteria for each of the principles are defined in detail in the *Revised Software Trust Principles* [BMD 1993]. Only if the compliance requirements for all of the principles are met for a given level can a development process can be classified as achieving that particular Trust Level.

A contractor that is following the TSM may use another person, the Trust Evaluator, to help ensure proper compliance to the trust requirements. A Trust Evaluator is independent from the development organization and provides TSM oversight throughout the entire development process. The software Independent Verification and Validation contractor may serve in this capacity or a separate contractor may be assigned this responsibility.

2.2 Overview of the SEI PMM

The Software Engineering Institute at Carnegie Mellon University was established by the Department of Defense (DoD) in 1984 to address software development issues that plague DoD's software-intensive systems. One of its tasks was to identify a means of evaluating a contractor's ability to effectively develop software. In June 1987, SEI defined an approach for determining the maturity level of a contractor's software development process [Humphrey 1987].

Table 1. Distribution of Trust Principles for Each Trust Level

Trust Level	Trust Principles
T5	Formal Methods
T4	Environment Integrity Reliability Engineering Intrusion Detection
T3	Shared Knowledge Distribution Testing Responsibility Access Control Testing Approach Auditing Code Analysis
T2	Security Identification & Authentication Computer-Assisted Software Engineering (CASE) Tools Standards Traceability Formal Review
T1	Reuse Prototyping Environment Administration Peer Review Documentation Configuration Management Risk Management Software Development Plan (SDP) Planning
T0	None

The underlying hypothesis on which this approach is based is that the quality of a software system is governed by the quality of the process used to develop and maintain it. The SEI methodology evaluates a contractor's software development and maintenance process as it is used on several projects, identifies the weaknesses of the process, and ranks its overall maturity.

The SEI model describes five maturity levels of an organization's software development and maintenance process which range from mature to immature. A mature process institutionalizes good software engineering techniques and is expected to produce software with reasonably consistent results, whereas an immature process lacks good software engineering practices and is expected to produce software with poor results (i.e., over budget and behind schedule). Following is a brief description of each maturity level [Humphrey 1989].

- Level 1 - Initial: The least mature organization is characterized as having "ad hoc" and "chaotic" processes. Since there are very few software engineering practices in place, it is very dependent on the people within the organization. It generally lacks good software project management, configuration management, quality assurance, and project planning practices.
- Level 2 - Repeatable: The organization has established basic project controls and is therefore thought to have a repeatable software development process in place. It is less dependent on individuals and has rigorous management oversight of commitments, change control, quality, and cost estimation.
- Level 3 - Defined: The organization has a foundation for defining the complete process and deciding how to improve it. Its process is more qualitative than quantitative in nature.
- Level 4 - Managed: The organization has a quantitative focus on their development process. The measurements extend beyond cost, schedule, and performance, and focus on quality and productivity.
- Level 5 - Optimizing: The organization is focused on continued improvement and optimization of the process.

The SEI maturity model, originally developed by Watts Humphrey in 1987, evolved around a questionnaire consisting of 110 questions [Humphrey 1987]. To determine a contractor's maturity level, an evaluation team would verify the contractor's responses to the questionnaire and apply a scoring algorithm. In 1989, SEI moved away from the scoring

algorithm and grouped the questions into eight KPAs, listed in Table 2. No formal name was assigned to this version of the model, but IDA refers to it as the Process Maturity Model (PMM). From 1989-1993, this was the version of the model taught in the SCE training course. Recently, however, SEI revised the model and expanded the evaluation criteria. The new model, commonly referred to as the Capability Maturity Model (CMM), is scheduled to be taught in the SCE training course in 1994. Until the CMM course is available and training material is updated, the SCE teams must use the PMM.

There are two methods for determining an organization's maturity level: the Software Capability Evaluation (SCE) and the Software Process Assessment (SPA). The SCE team consists of at least four software development experts trained by SEI. Each site visit takes approximately three days, interviewing software personnel and reviewing the contractor's software documents. For example, when exploring the contractor's configuration management process, the evaluation team interviews configuration managers and software developers to understand their process for making changes to software designs, code, and test cases. To substantiate answers to the interview questions, the evaluation team reviews plans and supporting documentation such as the Configuration Management Plan and minutes from a recent Configuration Change Board meeting. The combination of the interviews and supporting documentation helps to ensure that the process documented is the one that is used.

The SPA team consists of 4 to 10 software development experts trained by a contractor licensed by SEI. The SPA takes approximately five days, conducting group interviews with Functional Area Representatives (e.g., Configuration Managers, Quality Assurance representatives). The group interviews provide a forum for the selected representatives to discuss the software development process. The SPA team helps to facilitate the identification of problems occurring within the software development process. In addition, the SPA team solicits input from the Functional Area Representatives and develops a process improvement plan to mitigate the weaknesses that were identified.

2.3 BMD Software Policy

BMDO Directive 3405 [BMDO 1993] establishes requirements for Mission Critical Computer Resources (MCCR) software development and support activities. The policy directs the BMD program segments, major defense acquisition programs, and elements to use both the TSM and the SEI PMM.

Table 2. Distribution of SEI KPAs

Process Maturity Level	KPAs
5 Optimizing	Process Improvement Defect Prevention
4 Managed	Process Measurement Quality Management
3 Defined	Software Engineering Process Group (SEPG) Training Peer Review Standards and Procedures
2 Repeatable	Quality Assurance (QA) Configuration Management (CM) Project Planning Project Management
1 Initial	N/A

The Directive requires each program manager to be responsible for determining the Trust Level to which software components will be developed [BMDO 1993]. The program manager must identify Trust Principles that are candidates for non-implementation based on program performance, support, cost, and schedule constraints. The program manager must also assess the risks of not complying with each Trust Principle, identify options to mitigate the risk of non-compliance, or accept the program risks associated with non-compliance. In addition, an independent Trust Evaluator must be designated to monitor, assess, and report the level of compliance with software trust requirements throughout the software life cycle.

The Directive also requires SCEs to be performed during the source selection process with the results being used as part of the evaluation criteria. In addition, periodic SCEs (a cycle of not less than one year or more than two years) shall be performed to measure and monitor contractor process improvement. Prime contractors and subcontractors are encouraged to perform annual SPAs and program managers are encouraged to monitor the contractor's software process improvement program.

Currently, BMDO Directive 3405 does not call for an independent government evaluation of the contractor's ability to adhere to the TSM, i.e., a TSM/PMM evaluation.

But if the methodology proposed in this paper is effective, it may be included in future versions of the Directive.

3. MODEL DESCRIPTION

This section provides an overview of the TSM/PMM model and evaluation criteria that combines the requirements from both the TSM and the PMM.

3.1 Overview of TSM/PMM Model

The recommended model for combining the TSM and the PMM is limited to the first three levels of trust and maturity. The SCEs conducted to date have all evaluated contractors relative to a level 3 on the process maturity scale. The initial TSM evaluations are also scheduled for contracts that require adherence to all the principles up to and including Trust Level 3 (T3). In the event an organization only has requirements for adhering to T2 criteria, the T3 criteria can be easily removed from the TSM/PMM model by the evaluation team.

Since BMDO intends to evaluate a contractor's development risks relative to level 3 for both of these models, the Venn diagram in Figure 1 identifies the areas of overlap and the differences between the two models for levels 1, 2, and 3. As illustrated, many of the PMM KPAs are similar to the Trust Principles. Only three of them are not included in the TSM: Quality Assurance, Training, and SEPG.

Table 3 identifies the key process areas at their associated levels and the inconsistencies that exist between the two models. Several of the SEI KPAs are similar to the Trust Principles but they appear at different levels in the two models. For example, **Configuration Management** appears at PMM level 2 but it appears at level 1 in the TSM. **Peer Review** appears at Level 3 on the maturity scale but at Level 1 in the TSM.

It would be very difficult for an evaluation team to use the TSM and PMM as currently defined. The team must identify the areas of overlap between the two in order to refrain from duplicating efforts and to perform efficient interviews and documentation reviews. For example, it would be inefficient to interview the Configuration Manager on the first day of the evaluation to address the SEI criteria and then to recall the Configuration Manager on the last day to address the TSM criteria. It is preferable to group all evaluation criteria pertaining to Configuration Management and to address all of them at one time. In

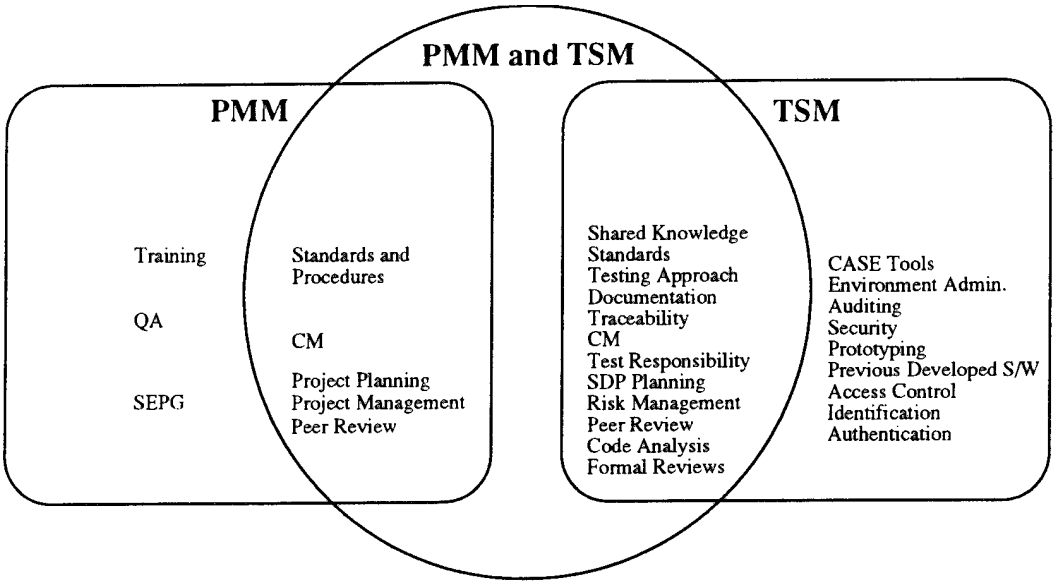


Figure 1. Overlap Between PMM and TSM

Table 3. A Comparison Between PMM KPAs (Levels 1-3) and TSM

Maturity Level	PMM KPA	Trust Level	Trust Principles
3	SEPG Training **Peer Review** Standards and Procedures	3	Shared Knowledge Distribution Testing Responsibility Access Control Testing Approach Auditing Code Analysis
2	Quality Assurance **Configuration Management** Project Planning Project Management	2	Security Identification & Authentication CASE Tools Standards Traceability Formal Reviews
1	N/A	1	Reuse Prototyping Environment Administration **Peer Review** Documentation **Configuration Management** Risk Management SDP Planning
		0	N/A

addition, it is cumbersome for the evaluation team to recall separate evaluation criteria for 8 KPAs and 25 Trust Principles for a total of 33 areas. Individuals are limited in their capacity to recall the details of such a large number of areas; thus it is preferable to condense the categories into a more manageable size.

To eliminate duplication between the two approaches and to streamline the criteria for the evaluation teams, a new model can be formed that combines the KPAs and Trust Principles that are similar and groups the unique Trust Principles into a smaller set of new Trust KPAs. Table 4 illustrates a revised model that consists primarily of 10 KPAs; 8 of these KPAs are from the PMM and 2 new KPAs were formed from unique TSM requirements. The KPAs include the following: Project Management, Project Planning, Configuration Management, Quality Assurance, Standards and Procedures, Peer Reviews, Training, SEPG, *S*ecure Development Environment (new), and Project Trust Policy (new).

Table 4. TSM/PMM Model

SEI KPA	Added Trust Principles Consistent with KPAs	New Trust KPAs	Trust Principles Unique to TSM
SEPG	(none)	Secure Development Environment	Access Control (T3) Auditing (T3) CASE tools (T2) Environment Administration (T1) Identification & Authentication (T2)
Training	(none)		
Peer Review	Shared Knowledge (T3) Peer Review (T1)		
Standards & Procedures	Testing Approach (T3) Standards (T2) Documentation (T1)		
Quality Assurance	(none)	Project Trust Policy	Distribution (T3) Security (T2) Prototyping (T1) Previous Developed Software (T1)
Configuration Management	Traceability (T2) CM (T1)		
Project Planning	Test Responsibility (T3) SDP Planning (T1)		
Project Management	Code Analysis (T3) Formal Reviews (T2) Risk Management (T1)		

3.2 Key Process Areas

This section includes a brief overview of the KPAs in the TSM/PMM model and provides the rationale for combining the various Trust Principles with the SEI KPAs.

a. SEPG: This KPA represents an organization within the contractor's company responsible for improving the software development process. Typical responsibilities of an SEPG include performing SPAs, developing organization development standards, maintaining the training data base, and collecting organization-wide measurements and costing data. The TSM does not have process improvement requirements.

b. Training: This KPA requires identifying the training requirements of projects and individuals, and developing or procuring the appropriate training courses to satisfy these requirements. The TSM does not have any principles associated with training.

c. Peer Review: This KPA is a method for identifying and removing defects as they appear in the software development products (e.g., requirements, design, code, and test cases) throughout the life cycle. Since both the Peer Review and the Shared Knowledge TSM principle require people to understand and identify errors in the software products, these principles were combined with the Peer Review KPA.

d. Standards and Procedures: This KPA accounts for the development and maintenance of the organization's standard software process. Since standards define the organization's approach for developing, testing, and documenting software, three trust principles were added to the Standards and Procedures KPA. The Testing Approach Principle identifies the method used to define, implement, and document test cases. The Standards principle defines what must be standardized (e.g., development methods, tools, products). And the Documentation Principle defines the requirements for adequately documenting the software life cycle products.

e. Quality Assurance: Quality assurance involves an independent organization whose responsibility is to review and audit software products to verify that they comply with the development plans, policies, and standards. There are no defined TSM principles currently associated with quality assurance functions.

f. Configuration Management: Configuration management systematically controls changes to the software system as it evolves and helps to maintain the traceability of the configuration throughout the life cycle. Both the Configura-

tion Management Principle and Traceability Principle are captured under this KPA.

g. Project Planning: This KPA involves establishing initial estimates and plans for implementing the software development effort. The two Trust Principles included in this KPA are the Project Planning Principle and Testing Responsibility Principle. The Project Planning Principle accounts for the development of the Software Development Plan (SDP) and the Testing Responsibility Principle accounts for the assignment of responsibility to the testing organization. Both of these principles involve activities addressed in the initial planning stages of the software life cycle.

h. Project Management: This KPA involves the activities associated with identifying and resolving problems once the software project is underway. The TSM Risk Management and Formal Review principles are combined with the Project Management KPA since they both offer a means of identifying project risks throughout the software life cycle. Since the Code Analysis principle involves measuring attributes of the software code to identify potential problem areas, it too was included in this KPA.

i. Secure Development Environment: This KPA includes the Trust Principles that pertain to the controls and capabilities of a project's software engineering environment (SEE). These added requirements for the SEE are unique to the TSM and therefore are not reflected in the SEI PMM.

j. Project Trust Policy: This KPA includes the additional procedures and practices necessary to maintain control of software development artifacts, including prototypes and previously developed software. These added trust policy requirements are unique to the TSM and therefore are not reflected in the SEI PMM.

3.3 Evaluation Criteria

Evaluation criteria have been defined for each of the KPAs in the TSM/PMM model. Criteria have two parts. The first part identifies the key practices previously defined in [SEI 1992a]. The second part identifies the key TSM practices defined in [BMD 1993] that have similar scope to the PMM practices.

Since each of the TSM principles has as many as 30 practices, an evaluation team will only have time to audit the most important ones. Therefore the TSM/PMM evaluation

criteria contain a condensed listing of the key practices defined in the TSM. Each practice is cross-referenced with the Trust Principles documented in [BMD 1993] so that the evaluation teams can supplement the key practices with other practices that are deemed important. Appendix A contains the detailed evaluation criteria for each of the TSM/PMM KPAs.

4. CONDUCTING EVALUATIONS

The typical SCE evaluation is done in a three-day site visit at the contractor's facility. The evaluation team conducts interviews and documentation reviews to assess the quality of the contractor's software development process. With the addition of the TSM to the SCE process, several changes must be made to the activities conducted prior to the site visit, during the site visit, and at the close of the site visit. The purpose of this section is to define those SCE activities affected from the addition of the TSM requirements.

4.1 Request for Proposal Requirements

When SCEs are used for source selection or contract monitoring, the contractors must be made aware of the TSM/PMM requirements in the Request for Proposal (RFP). Appendix B of this document provides text that can be used in the RFP to state that a TSM/PMM evaluation will be performed. It gives the appropriate references, informing the contractor of the criteria that will be used to evaluate its software development process. The text is expected to be tailored to accommodate the specific requirements of a BMD element acquisition program.

4.2 Questionnaire

As part of the SCE process, the contractor completes the PMM questionnaire and returns it to the evaluation team prior to the site visit. The PMM questionnaire consists of approximately 85 Yes/No questions which help to characterize the contractor's software development process. The evaluation team uses the questionnaire to perform a preliminary analysis of the contractor's process and to identify potential weaknesses to explore during the site visit. Since the PMM questionnaire does not account for all of the Trust Principles, the SEI questionnaire was extended. Refer to Appendix C for a copy of the revised questionnaire that will be used when applying the TSM/PMM model.

4.3 Document Requests

The evaluation team can request project documentation during different stages of the site visit. Prior to the SCE site visit, the evaluation team will have requested and

received documents that help it become acquainted with the contractor's process (e.g., the SDP). The SDP should contain specific references on how the contractor plans to satisfy the Trust Principles defined in the TSM. The SDP should help the evaluation team become acquainted with the contractor's trusted software process prior to the site visit.

Once the evaluation team goes on site, additional documents will be requested and reviewed in order to substantiate the contractor's development process. Appendix D contains a list of documents that should be reviewed for each of the TSM/PMM KPAs.

4.4 Interview Schedule

With the addition of the TSM criteria, it is expected that the site visit will take approximately four days to complete. The additional time is spent conducting the necessary interviews and documentation reviews associated with the Trust Principles. Table 5 represents a scheme for allocating time for interviews during the four-day site visit.

Table 5. Allotted Time for Interviews

Position	Length of Interview (hrs)	Number Interviewed
VP of Software	0.50	1
Project Managers	0.50	2
S/W Managers	1.25	2
Manager of QA	0.50	1
Project QA	0.75	1
Manager of CM	0.50	1
Project CM	0.75	1
SEPG	1.25	1
Standards	0.75	1
Training	0.50	1
S/W Cost Manager	0.50	1
Subcontract Manager	0.75	1
Subcontractor	0.75	1
Developer	1.25	1
Chief of Computer Security	1.25	1
SEE Administrator	1.25	1

Over 18 individuals will be interviewed during the four-day evaluation period. The amount of time allocated to interview an individual is proportional to the number of KPAs typically under the responsibility of that individual and the individual's detailed knowledge of the process area. Table 6 identifies the KPAs that will be reviewed during each of these interviews.

Table 6. Subject of Interviews

Title of Interview Candidates	TSM/PMM KPAs									
	PM	PP	CM	QA	SP	PR	TR	SEPG	EV	TP
VP of Software	X	X								
Project Manager	X	X		X	X					X
Software Manager	X	X			X	X		X		X
CM Manager		X	X		X				X	X
QA Manager		X		X	X	X				
CM Engineer			X		X					
QA Engineer				X	X	X				
Developer	X		X			X	X		X	X
SEPG Manager						X		X		
Training Manager							X			
Chief of Computer Security			X						X	X
SEE Administrator			X						X	X
S/W Cost Manager		X								
Subcontract Manager	X									X

The SCE team should use organization charts to identify the titles and names of the individuals with their assigned responsibilities. In some cases, candidates may be responsible for several KPAs. Thus, appropriate adjustments should be made to the length of the candidate's interviews, depending on the assigned responsibilities.

Using the allocations specified in Tables 5 and 6, the site visit should follow the daily schedules described in Figures 2 through 5. These schedules allow 15-minute breaks between most interviews and lunch around noon each day. Figure 2 provides the sample schedule for the first day of the site visit, beginning with an entrance briefing from both the evaluation team and the contractor. The remainder of the first day is spent conducting interviews with senior project personnel. The second day of the site visit involves interviewing personnel responsible for specific KPAs such as configuration management and quality assurance personnel. The third day is primarily devoted to exploring the leading Trust Principles such as the Project Trust Policies and the Secure Development Environment. The fourth day is devoted to performing additional interviews as needed, preparing the findings, and presenting the TSM/PMM results.

8:30 - 9:00	BMDO Introductory Brief
9:00 - 10:00	Contractor Entrance Briefing
10:00 - 11:00	Documentation Review
11:00 - 5:30	Interviews:
	1.25 hrs S/W Lead (project A)
	0.25 hrs Break
	0.50 hrs Project Manager (project A)
	0.25hrs Break
	0.75 hrs SEPG Manager
	0.25 hrs Break
	1.25 hrs S/W Manager (project B)
	0.25 hrs Break
	0.75 hrs S/W QA

Note: Lunch around 12:00 (1hr)

Figure 2. Schedule for Day One

8:30 - 10:30	Interviews:
	0.75 hr Standards & Training
	0.50 hr S/W Costing Manager
	0.75 hr S/W Configuration Management
10:30 - 1:30	Documentation Review and lunch
1:30 - 5:00	Interviews:
	0.75 hr CM Engineer
	0.25 hr Break
	0.50 hr QA Manager
	0.50 hr Project Manager (project B)
	0.25 Break
	1.25 hr S/W Developer

Note: Lunch around 12:00 (1 hr)

Figure 3. Schedule for Day Two

8:30-10:45	Interviews:
	1.25 hrs Chief of Computer Security
	0.25 hrs Break
	1.25 hrs SEE Administrator
10:45-1:30	Documentation Review and lunch
1:30-3:30	Interviews:
	1.25 hrs Subcontractor Manager
	0.25 hrs Break
	0.50 hrs VP of Software
3:30-5:00	Additional interviews (TBD)
7:30-9:30	Documentation Review (at hotel)

Note: Lunch around 12:00 (1hr)

Figure 4. Schedule for Day Three

8:00 - 9:00	Documentation Review
9:00 - 12:00	Interviews as needed
12:00 - 3:30	Prepare Exit Briefing (or Report)
3:30 - 5:00	Exit Briefing

Figure 5. Schedule for Day Four

4.5 Trust Evaluator

A Trust Evaluator is a person or organization independent of the BMD contractor. The Trust Evaluator is typically on site at the contractor's facility and responsible for overseeing that the contractor implements the TSM requirements. The evaluator spends substantial time (e.g., 0.5 labor year) reviewing and monitoring the contractor's trust practices.

The role of the Trust Evaluator during a TSM/PMM evaluation can vary, depending on the purpose of the evaluation. Evaluations can be performed for source selection or to help monitor a contract after it is awarded. Contractors competing for a new BMD program during the source selection process will generally not have a Trust Evaluator since this is a unique requirement of the BMD program. However, contractors that are already under contract with BMDO will likely have a Trust Evaluator on board.

When an TSM/PMM evaluation is being used to help monitor a contract, it would be beneficial for the SCE evaluation team to leverage from the information the Trust Evaluator has collected. It is recommended that the SCE team conduct approximately 75% of the evaluation and then question the Trust Evaluator only on the areas where weaknesses were found. The team should perform essentially an independent evaluation and meet with the Trust Evaluator to identify information which the team may have overlooked. Based on the Trust Evaluator's input, the team will have an opportunity during the remainder of the site visit to verify the input received from the Trust Evaluator through additional interviews and documentation reviews.

5. RESULTS OF TSM/PMM EVALUATION

This section of the report includes the format of the evaluation results.

5.1 KPA Findings

The TSM/PMM evaluation findings are similar to the SCE findings that identify the strengths and weaknesses associated with each KPA. However, the TSM results will be identified separately from the PMM results for each KPA. Figure 6 provides a sample of the results that may be found for the KPA entitled Peer Review. As illustrated, the contractor had strengths and weaknesses associated with both the PMM criteria and the TSM criteria. Since the TSM criteria are generally more stringent than the PMM criteria, it is feasible that the contractor's process may satisfy the PMM requirements but not the TSM requirements. For this reason, it is important to keep the findings distinct.

5.2 Summary of Evaluation Results

After the detailed findings for each KPA are defined, the evaluation team will produce a summary of the overall findings, as illustrated in Table 7.

KPAs shaded in Table 7 mean only one of the models has requirements associated with these KPAs. For example, the TSM does not include requirements for SEPG or Training. Similarly, the PMM does not contain requirements for the Secure Development Environment or the Project Trust Policy.

The evaluation team identifies the limits of the contractor's process for the Source Selection Evaluation Board (SSEB) and the Government program office. But in general, the evaluation team does not assign a Maturity Level score (e.g., level 1, 2, or 3) or a Trust Level score (T1, T2, or T3) to the contractor's process.

Peer Review

General SCE Results: *Acceptable* peer review process at Maturity Level 3

Strengths:

- Organization procedures define the review process.
- Training is provided for moderator and peer reviewers.
- Review findings are maintained and tracked.

Weaknesses:

- Review schedule and assignments not published and distributed.

General TSM Results: *Unacceptable* peer review process at Trust Level 3

Strengths: (same as SEI's above plus)

- Peer review teams include three people (author, outside reviewers).

Weaknesses: (same as SEI's above plus)

- Peer review team does not consistently include four people (T3).
- No method to ensure items under review are under configuration control (T2).
- Two people are not consistently assigned to share knowledge and responsibility for all identified SEE components (T3).

Figure 6. Sample KPA Findings

Table 7. Summary of KPA Results

KPAs	PMM Requirements (L1 - L3)		TSM Requirements (T1 - T3)	
	Acceptable	Unacceptable	Acceptable	Unacceptable
SEPG	X			
Training	X			
Peer Review	X			X
Standards		X		X
Quality Assurance	X			
Configuration Management		X		X
Project Plan	X		X	
Project Management	X		X	
Secure Development Environment				X
Project Trust Policy			X	

APPENDIX A.
TSM/PMM EVALUATION CRITERIA

Note: The "Additional TSM Criteria" are labeled with the appropriate references to the Trust Level (T1 - T5) and the compliance requirements (a - z) listed in [BMD 1993].

Project Management Evaluation Criteria

1. SEI's Key Practices

- Software management helps to develop, review, and commit to software size, schedules, and budget estimates. The commitment process is defined, documented, tracked, and enforced.
- Project responsibilities are defined and documented, particularly the relationship between software and systems engineering, software requirements team and design team, and responsibilities associated with each milestone and deliverable.
- Process exists for raising, tracking, and resolving issues.
- Senior management is regularly briefed on the status of software development during the course of development.
- Software risks associated with technical activities, cost, schedule, and resources are identified, assessed, and documented.
- Subcontractors are selected and monitored by a documented process involving the prime contractor's software organization, and the subcontractor's development efforts comply with the prime contractor's standards and procedures.
- A subcontractor's Software Development Plan (SDP) is approved by the prime contractor and used to track the subcontractor's progress.

2. Additional TSM Criteria

2.1 Code Analysis Principle (T3)

- The following metrics shall be collected and analyzed (c):
  - Completeness: All requirements are implemented.
  - Modularity: High cohesion and optimum coupling are achieved.
  - Simplicity: Code implementation is non-complex.
- Code analysis shall be performed to identify unused code which shall be either (1) removed prior to delivery or (2) identified as a risk item in the SDP if delivered (e).

2.2 Formal Reviews Principle (T2)

- DOD-STD-2167A reviews shall be conducted on all delivered configuration items (a).

- All action items resulting from the formal reviews shall be documented, assigned, and monitored, and a status report shall be delivered on a regular basis (g).
- All material under review shall be under configuration management (j).

2.3 Risk Management Principle (T1)

- The SDP shall include a description of the risk management techniques (e.g., identification, analysis, prioritization, monitoring, and resolution) (a).
- Risk monitoring shall be performed continually throughout the life cycle (g).

Project Planning Evaluation Criteria

1. SEI's Key Practices

- Estimated size, cost, and schedule of the software development effort are based on past performance data (calibrated cost model, data base).
- Policy and procedures exist for developing software estimates.
- Software managers are trained on the estimation techniques.
- Metrics are used to track actual versus planned development progress, code and test errors, critical computer resources (utilization, capacity, throughput).
- Estimates are monitored regularly and updated when affected by change.

2. Additional TSM Criteria

2.1 Testing Responsibility Principle (T3)

- Computer software component (CSC) and computer software configuration item (CSCI) testing responsibility shall be separate from developers (a).
- Test developers shall understand requirements (i.e., purpose, source, traceability) before developing or executing tests (c).

2.2 SDP Planning Principle (T1)

- The SDP shall identify methods of satisfying Trust Principles in sufficient detail to allow the verification of compliance (d).
- The SDP shall identify software trust requirements traceability (e).
- No deviations from the software development approach documented in the SDP shall be made without prior approval from a designated authority (i).
- All software development personnel shall be familiar with the SDP methods (k).

Configuration Management Evaluation Criteria

1. SEI's Key Practices

- Project-level configuration management (CM) plans are prepared according to organization procedures.
- Change control board is responsible for various baselines of the development product (requirements, design, code, test, plans, procedures, interfaces).
- Documented change control process exists (CM plan, forms for reporting errors, program library system, configuration management tools, check in/out procedures, configuration status reporting).
- Regression tests regularly conducted to ensure changes have not introduced new errors (adequacy and frequency of regression testing is defined).
- Configuration status report exists to identify status at any point in time and status of open/closed change requests.
- Forward and backward traceability established for requirements, design, code, and tests.
- Development baseline is under configuration control after unit test.

2. Additional TSM Criteria

2.1 Traceability Principle (T2)

- Forward and backward traceability is established between system requirements, software requirements, design, code, and CSCI, CSC, and computer software unit (CSU) tests. Traceability is also established between software requirements and CSCI and CSC test cases (a - k).
- Traceability is maintained for all new software requirements and for all modifications to existing software requirements (l).

2.2 Configuration Management Principle

- The CM system shall enforce multiple levels of control formality commensurate with the criticality of the item (e.g., accessibility, approval) (f) (T2).
- CM system shall include procedures for handling hard copy items under configuration control (l) (T1).

- Controls shall be in place to detect and prevent attempts to perform unauthorized modifications to items stored in the development library (u) and to the CM system itself (a, f) (T2).
- It shall be possible to regenerate any version of an item under configuration control (y) (T1).
- A mechanism shall be incorporated into the CM system to allow for comparison of software versions to ensure only authorized modifications have been performed (z) (T3).
- The CM system shall record the user identification, date, and time of an operation that updates an item under configuration control (a, c) (T3).

Quality Assurance Evaluation Criteria

1. SEI's Key Practices

- Quality assurance (QA) has a separate reporting chain to senior management.
- Project-level QA plans are prepared according to documented organization procedures.
- QA has authority (stop work authority at any time, required for major transitions in development).
- QA audits the development products (reviews all line activities, products at all phases of the life cycle, audits subcontractor's products).
- Audit process and procedures are well defined (established representative sampling technique, problem tracking and reporting).
- Sufficient resources are assigned to QA (3 to 5%).

2. Additional TSM Criteria

- None.

Standards and Procedures Evaluation Criteria

1. SEI's Key Practices

- Standards exist for SDP, QA plan, CM plan, coding, unit development folders, and man-machine interfaces.
- Responsibility has been for development and maintenance of standards.
- Audit criteria have been established.
- Management is committed to use of standards.

2. Additional TSM Criteria

2.1 Testing Approach Principle (T3)

- All test cases (CSU, CSC, and CSCI) identify requirements that will be verified, inputs required to execute the test case, expected results, and dependencies on other test cases (b).
- CSU test cases include minimum/maximum input/output boundaries, illegal values, and 100% branch and statement testing (c).
- There is a method to ensure tested software is under configuration control (l).
- All test cases shall be designed to ensure all allocated requirements are satisfied (d).

2.2 Standards Principle (same as SEI's) (T2)

- Software standards shall be specific to the methods, tools, languages (c) and software products (requirements, design, code, tests) (a).
- There is a procedure to enforce standards (e).
- Standards are updated when inconsistencies are identified (d).

2.3 Documentation Principle (T1)

- Each software deliverable shall be adequately documented, i.e., software requirements (c), designs (e), code (f), test plan (g).

Peer Review Evaluation Criteria

1. SEI's Key Practices

- SDP identifies peer reviews during specific phases (design, code, test).
- Technical review schedule is published periodically and distributed widely.
- Review assignments are published.
- A process is defined and documented (preparation, conduct, reporting).
- Review findings are maintained and tracked.
- Statistics are gathered on the conduct of the peer review and product errors to improve the peer review process and the development process.

2. Additional TSM Criteria

2.1 Shared Knowledge Principle (T3)

- There shall always be at least two people who share knowledge and responsibility for installation, configuration, and operation of a SEE component (a).
- Items requiring shared knowledge are logged (assignments, responsibilities) (c).
- The personnel assigned to share knowledge of an item shall have (d):
  - Equal responsibility for correctness and completeness of the item.
  - Comparable knowledge of the assumptions, alternatives, and critical decisions during creation and maintenance.
  - Comparable knowledge of tools, methods, languages, and procedures necessary to maintain the item.

2.2 Peer Review Principle

- There is a method to ensure items under review are under configuration control (p) (T2).
- Peer review moderator is trained to conduct effective reviews (o) (T3).

Training Evaluation Criteria

1. SEI's Key Practices

- Training requirements are established for each job function (configuration manager, quality assurance personnel, peer review moderators, project managers, software supervisors, software developers).
- Training policy and resources are established (money, facilities, tools, schedules, and waiver procedures).
- Training records are maintained identifying who has been trained for each course.
- Organization training plan is established identifying current and future course offerings and needs.
- Project training plan is established identifying its training needs.

2. Additional TSM Criteria

- None.

Software Engineering Process Group (SEPG) Evaluation Criteria

1. SEI's Key Practices

- Full-time resources are assigned to define and improve the organization's software development process.
- A central resource is established for software engineering tools, training plans, organization metrics, process model, and lessons learned.
- Identifiable improvement activities have been accomplished and realistic plans for further improvement have been established.
- SEPG maintains visibility of software projects' process and technology needs and receives project input to establish future SEPG improvement efforts.
- SEPG has a mechanism to transfer new technology to the projects.
- The organization's software process is assessed periodically and action plans are developed to improve the process (e.g., software process assessment).

2. Additional TSM Criteria

- None.

Secure Development Environment Evaluation Criteria

1. SEI's Key Practices

- None.

2. Additional TSM Criteria

2.1 Access Control Principle (T3)

- Access control shall be implemented by the SEE (a).
- Discretionary Access Control (DAC) mechanisms shall be employed (c).
- Discretionary access permission shall be set by an authorized user in accordance with security policy (e).

2.2 Auditing Principle (T3)

- An audit trail shall be automatically logged by the SEE for the following activities: deletion of controlled copies in Software Development Library, use of identification and authentication, and attempts to access resources without authorization (a).
- Audit trail can reconstruct security violations or malfunction (b).
- Audit mechanism shall be tamper proofed and the audit trail repository protected (c).

2.3 Identification and Authentication Principle (T2)

- Identification and authentication shall include the sequence of events in which a user's identity is established and verified before accessing the SEE (a).
- Identification and authentication mechanisms shall be protected (r).
- Policy and procedures ensure authentication can not be easily guessed (e).
- All accounts shall be reviewed regularly to verify access is appropriate (l).
- The SEE has ability to automatically expire a user's authentication data after a specified period to enforce the regulation on users changing passwords regularly (o) (T3).
- The SEE shall identify a user and maintain a record of privileges (c) (T3).

2.4 CASE Tools Principle (T2)

- CASE tools shall verifying compliance with project standards (c).

- CASE tools shall be used for requirements analysis, design (d), coding (e), and testing (f).

2.5 Environment Administration Principle (T2)

- Administrative procedures for SEE components shall be documented (e.g., installation, configuration, operation, maintenance) (a).
- Administrative procedures for SEE also include file recovery (g), disaster recovery (h), and procedures for controlling modifications to the SEE (i).

Project Trust Policy Evaluation Criteria

1. SEI's Key Practices

- None.

2. Additional TSM Criteria

2.1 Distribution Principle (T3)

- Software shall be transferred in a manner that protects it from tampering and allows authentication upon receipt (a).
- Administration is coordinated with sending and receiving organizations (d).

2.2 Security Principle (T2)

- A security policy is defined, reviewed, and endorsed by software management (a).
- Conditions are defined for identifying SEE users (f).
- Conditions are defined to ensure people using SEE are accountable (l).
- Conditions are defined to evaluate whether SEE enforces the policy (n).
- Procedures are established for reporting, investigating, and rectifying violations (v).
- There is a training program to ensure all understand security policy (w).

2.3 Prototypes Principle (T1)

- Proof-of-concept prototypes and their artifacts shall not be reused in deliverable software products (d).
- All developmental prototypes shall be developed in accordance with trust requirements (e).

2.4 Reuse Principle (T1)

- A description of each previously developed software item shall be documented (i.e., use, rational for use, assessment of risk, risk mitigation approach) (b).
- Reuse risk assessment is based on documented criteria (i.e., suitability, Trust Level, origin and history, supportability, criticality) (c).

APPENDIX B.
TSM/PMM TEXT FOR INCLUSION IN THE REQUEST FOR PROPOSAL

The following sample text illustrates how the TSM/PMM evaluation may be inserted within Section L or M of the Request for Proposal. This example assumes that the Software Capability Evaluation (SCE) will be used as a specific criterion for source selection.

Software Engineering Capability. The Government will evaluate the software process by reviewing the offeror's Software Process Improvement Plan, by using the Software Engineering Institute (SEI) developed technique, Software Capability Evaluation (SCE), and by applying the BMDO Trusted Software Methodology (TSM). The Government will determine the software process capability by investigating the offeror's current strengths and weaknesses in key process areas defined in the SEI report, *Characterizing the Software Process: A Maturity Framework*, CMU/SEI-87-TR-11, and Trust Principles defined in the *BMD Software Standards Document*. The Government will perform an SCE of each offeror by reviewing current projects at the site proposing on this contract. The evaluation will be an organizational composite, substantiated through individual interviews and reviews of documentation, of the offeror's software process practices on selected projects. The evaluation will determine the offeror's strengths and weaknesses in key process areas relative to Maturity Level three and Trust Level three, i.e., the extent to which an offeror meets or exceeds Maturity Level or Trust Level three criteria. The on-site evaluators may be separate and distinct from the proposal evaluation team and may include a Government contracting representative. The evaluators will have received SCE and TSM training.

APPENDIX C.
TSM/PMM QUESTIONNAIRE AND RESPONSE FORM

This form is a modified version of the Software Engineering Institute (SEI) Process Maturity Model (PMM) questionnaire. It should be referenced and included in the Request for Proposal, and filled out by each contractor.

Name of Projects

Project A: _____

Project B: _____

Project C: _____

Project D: _____

Project E: _____

Project F: _____

Project G: _____

Project H: _____

Project I: _____

Note: The numbered and asterisk questions were taken from [Humphrey 1987]. The questions without numbers were derived from [BMD 1993].

	A	B	C	D	E	F	G	H	I
PROJECT MANAGEMENT									
2.1.3* Is a *formal procedure* used in the management review of each software development prior to making contractual commitments?									
2.1.4 Is a *formal procedure* used to assure periodic management review of the status of each software development project?									
2.4.1* Does senior management have a *mechanism* for the regular review of the status of software development projects?									
2.4.7* Do software development first-line managers sign off on their schedules and cost estimates?									
1.1.1 For each project involving software development, is there a designated software manager?									
1.1.2 Does the project software manager report directly to the project (or project development) manager?									
2.4.4 Is a *mechanism* used for independently calling integration and test issues to the attention of the project manager?									
2.1.17 Is a *mechanism* used for ensuring that the software design teams understand each software requirement?									
2.4.3 Is a *mechanism* used for identifying and resolving system engineering issues that affect software?									
1.1.5 Is software system engineering represented on the system design team?									
2.4.10 Is there a formal management *process* for determining if the prototyping of software functions is an appropriate part of design *process*?									

	A	B	C	D	E	F	G	H	I
2.4.5 Is a *mechanism* used for regular technical interchanges with the customer?									
2.1.5 Is there a *mechanism* for assuring that software subcontractors, if any, follow a disciplined software development *process*?									
Is there a mechanism for reviewing code metrics as a basis for targeting reviews, rework, and testing efforts?									
Is there a procedure for documenting, assigning, tracking, and closing action items resulting from formal reviews?									
Is there a process for identifying and monitoring risks throughout the software life cycles?									
Is there a trust evaluator?									
2.1.14* Is a *formal procedure* used to make estimates of software size?									
2.1.16* Are *formal procedures* applied to estimating software development cost?									
2.2.7 Are profiles maintained of actual versus planned software units designed, over time?									
2.2.8 Are profiles maintained of actual versus planned software units completing unit testing, over time?									
2.2.9 Are profiles maintained of actual versus planned software units integrated, over time?									
2.2.18 Is test progress tracked by deliverable software component and compared to the plan?									
2.2.19 Are profiles maintained of software build/release content versus time?									

	A	B	C	D	E	F	G	H	I
2.1.15* Is a *formal procedure* used to produce software development schedules?									
2.2.1* Are software staffing profiles maintained of actual staffing versus planned staffing?									
2.2.2* Are profiles of software size maintained for each software configuration item, over time?									
2.2.10 Are target computer memory utilization estimates and actuals tracked?									
2.2.11 Are target computer throughput utilization estimates and actuals tracked?									
2.2.12 Is target computer I/O channel utilization tracked?									
Is there a means for ensuring that software testers are not assigned to areas for which they have had development responsibilities?									
Is there method to assure that test developers and testers are knowledgeable about the requirements of the products they test?									
Are the methods for complying to Trust Principles documented as part of the Software Development Plan?									
Is there a procedure for getting an official release to deviate from standard, documented software development methods and approaches?									
QUALITY ASSURANCE									
1.1.3* Does the Quality Assurance (QA) function have a management reporting channel separate from the software development project management?									
2.1.7 For each project, are independent audits conducted for each step of the software development *process*?									

	A	B	C	D	E	F	G	H	I
2.4.19* Is a *mechanism* used for verifying that samples examined by QA are truly representative of the work performed?									
2.4.6* Is a *mechanism* used for ensuring compliance with the software engineering *standards*?									
CONFIGURATION MANAGEMENT									
1.1.6* Is there a software configuration control function for each project that involves software development?									
2.4.9* Is a *mechanism* used for controlling changes to software requirements?									
2.4.17* Is a *mechanism* used for controlling changes to the code? (Who can make changes and under what circumstances?)									
2.4.13* Is a *mechanism* used for controlling changes to the software design?									
1.1.4 Is there a designated individual or team responsible for the control of software interfaces?									
2.4.8 Is a *mechanism* used for ensuring traceability between the software requirements and top-level design?									
2.4.11 Is a *mechanism* used for ensuring traceability between the software top-level and detailed designs?									
2.4.14 Is a *mechanism* used for ensuring traceability between the software detailed design and the code?									

	A	B	C	D	E	F	G	H	I
2.4.18 Is a *mechanism* used for configuration management of the software tools used in the development *process*?									
2.4.20 Is there a *mechanism* for assuring that regression testing is routinely performed?									
2.4.21* Is there a *mechanism* for assuring the adequacy of regression testing?									
Is there a procedure for maintaining traceability between requirements, design, code, and tests as requirements are added or modified?									
Is an automatic record kept of all operations that change an item under configuration control, including user identification, date and time, and operation?									
Is there a system to detect and prevent unauthorized changes to a controlled item?									
Is there a tool or mechanism for comparing two versions of a controlled item and for ensuring only authorized modifications have been performed?									
PEER REVIEW									
2.4.12* Are internal software design reviews conducted?									
2.4.16* Are software code reviews conducted?									
2.4.22 Are formal test case reviews conducted?									
2.2.13* Are design and code *review coverages* measured and recorded?									
2.2.4* Are statistics on software code and test errors gathered?									
2.2.3* Are statistics on software design errors gathered?									

	A	B	C	D	E	F	G	H	I
2.3.2* Are the *review data* gathered during design reviews analyzed?									
2.3.8* Is *review efficiency* analyzed for each project?									
2.2.16 Are software trouble reports resulting from testing traced to closure?									
2.2.15* Are the action items resulting from design reviews tracked to closure?									
2.2.17* Are the action items resulting from code reviews tracked to closure?									
Is there a method to ensure items under review are under configuration control?									
Is there a mechanism to ensure that peer reviews are held for all requirements, design, code, and test deliverables?									
For each software deliverable deemed to be "critical," are there always at least two people who understand it well enough to continue its development and maintenance?									
Are shared knowledge responsibilities documented?									
TESTING									
2.2.14* Is *test coverage* measured and recorded for each phase of functional testing?									
TRAINING									
1.2.2 Is there a required training program for all newly appointed development managers designed to familiarize them with software project management?									
1.2.4 Is there a required software engineering training program for first-line supervisors of software development?									

	A	B	C	D	E	F	G	H	I
1.2.5* Is a formal training program required for design and code *review leaders*?									
1.2.3* Is there a required software engineering training program for software developers?									
STANDARDS AND PROCEDURES									
2.1.9 Are coding *standards* applied to each software development project?									
2.1.6 Are *standards* applied to each software development project?									
2.1.11 Are code maintainability *standards* applied?									
2.1.10 Are *standards* applied to the preparation of unit test cases?									
2.1.18 Are man-machine interface *standards* applied to each appropriate software development project?									
2.1.12 Are internal design review *standards* applied?									
2.1.13* Are code review *standards* applied?									
Are there standards that define the format of software requirement, design, code, and test plan deliverables?									
Is there a procedure to enforce standards?									
Is there a mechanism to ensure that test cases cover min/max input/output values, illegal values, and 100% of branches and statements?									
Do standards for test plan deliverables specify that each test case includes a list of requirements that will be verified, inputs needed for the test case, expected results of test case, and dependencies on other test cases?									

	A	B	C	D	E	F	G	H	I
SOFTWARE ENGINEERING PROCESS GROUP									
1.1.7* Is there a software engineering *process group* function?									
2.1.1* Does the software organization use a standardized and documented software development *process* on each project?									
2.1.2 Does the standard software development *process* documentation describe the use of tools and techniques?									
2.4.15 Are formal records maintained of unit (module) development progress?									
2.3.1* Has a managed and controlled *process database* been established for *process metrics* data across all projects?									
2.3.9 Is the software productivity analyzed for major *process* steps?									
2.2.5* Are design errors projected and compared to actuals?									
2.2.6* Are code and test errors projected and compared to actuals?									
2.3.3* Is the error data from code reviews and tests analyzed to determine the likely distribution and characteristics of the errors remaining in the product?									
SECURE DEVELOPMENT ENVIRONMENT									
Does the SEE contain an automated access control mechanism?									

	A	B	C	D	E	F	G	H	I
Are discretionary access permissions set in accordance with a security policy?									
Does the SEE automatically create an audit trail for the following activities: deletion of controlled objects in the software development library, establishment and verification of a user's identity, and attempts to access resources without authorization?									
Are auditing mechanisms tamper proofed and are audit trails stored in a protected repository?									
Are identification and authentication mechanisms tamper proofed?									
Are there policies to ensure that authentication data cannot be easily guessed?									
Are users identified and authenticated before access to the SEE is granted?									
Are CASE tools used for requirements analysis and design?									
Are CASE tools used to support testing?									
Is there documentation describing the installation, configuration, operation, and maintenance procedures for SEE components?									
Is there documentation describing file recovery, disaster recovery, and modification control procedures for the SEE?									
PROJECT TRUST POLICY									
Is there a method for delivering software that protects it from being tampered?									
Is there a method for verifying that delivered software has not been corrupted?									
Is there a defined security policy that is endorsed by software management?									
Does security policy address the identification and accountability of SEE users?									
Are there procedures for reporting, investigating, and rectifying violations of the security policy?									
Is there a procedure for ensuring that proof-of-concept prototypes are not reused in deliverable products?									

	A	B	C	D	E	F	G	H	I
Are there procedures for ensuring that developmental prototypes are developed in accordance with standard software development practices?									
For each reusable software item, is there documentation describing the use, rationale for use, assessment of risk, and risk mitigation approach?									
Is the risk assessment for reusable software items based on documented criteria such as suitability, trust level, origin and history, supportability, and criticality?									
OTHER KEY PROCESS AREAS									
2.4.2* Is a *mechanism* used for periodically assessing the software engineering *process* and implementing indicated improvements?									
2.3.4* Are analyses of *errors* conducted to determine their *process* related causes?									
2.3.5* Is a *mechanism* used for error cause analysis?									
2.3.6* Are the error causes reviewed to determine the *process* changes required to prevent them?									
2.3.7* Is a *mechanism* used for initiating error prevention actions?									

APPENDIX D.
TSM/PMM DOCUMENTATION LIST

Project Management KPA

1. **Project Management**
    a. Policies and procedures for project management
    b. Monthly status reports
    c. Status reports of software risks (monthly, quarterly)
    d. Formal review reports, issues, action items
    e. Procedure for modifying the Software Development Plan (SDP)
    f. Policy and procedure directives on selecting and monitoring subcontractors
    g. Commitment process document for software size, schedule, and budget estimating
    h. Issue resolution process
    i. Minutes of an issue resolution meeting
    j. Subcontractor SDPs
    k. Policy and procedure for project resource expenditure management
2. **Code Analysis**
    a. SDP, in particular, the portions dealing with the identification of code attributes to be measured, the method for collecting and analyzing metrics data, and the process for identifying high risk code
    b. Code analysis report
    c. Code risk identification and mitigation report
3. **Formal Reviews**
    a. Formal review standards
    b. Formal review checklists
    c. Minutes of a formal review meeting, including action items
    d. Formal review action item status reports
4. **Risk Management Principle**
    a. Procedures for risk identification, assessment, and mitigation

b. Reports on identification and status of risks and risk mitigation activities

c. Risk Management Plan (part of SDP)

Project Planning KPA

1. **Project Planning**
    a. Procedure for estimating size, schedule, cost
    b. Procedure for collecting and reporting metrics (critical computer resources, staffing profiles, code/test errors, units completed testing and integration)
    c. SDP, which includes life cycle model, procedures and methods to be used in developing software, identification of software work products, size estimates for software work products, estimates of project effort and cost, estimates of computer resource needs, project schedules, software risk assessment, and plans for facilities and tools
    d. Policies and procedures for developing project level documents (SDP, Software Quality Assurance Plan, Software Configuration Management Plan, Software Testing Plan)
    e. Staffing plan
    f. Procedures for identification, assessment, and mitigation of project risks
    g. Risk Management Plan (may be part of the SDP)

2. **Testing Responsibility Principle**
    a. List of people assigned to develop and execute functional tests for each computer software component (CSC) and computer software configuration item (CSCI), in conjunction with list of people assigned to design and code each CSC and CSCI

3. **SDP Planning Principle**
    a. SDP

Configuration Management KPA

1. Configuration Management

a. Policies and procedures for configuration management

b. Software Configuration Management Plan

c. Procedure for change control

d. List of Configuration Control Board (CCB) members

e. CCB agendas

f. CCB minutes

g. CCB checklists

h. CCB submission forms

i. CCB system release notes

j. Error reporting form, error tracking form

k. Regression test report

l. Configuration status report

m. Policy and procedure directives on traceability

2. Traceability Principle

a. Documentation showing traceability of each software requirement to a system requirement, and of each system requirement assigned to a software requirement

b. Documentation showing traceability of each design component to a software requirement and of each software requirement to design components

c. Documentation showing traceability of each source code component to one or more design components and of each design component to one or more source code components

d. Documentation showing traceability of each source code component to one or more computer software units (CSU), CSC, and CSCI tests that exercise the source code component

e. Documentation showing traceability of each CSC and CSCI test to one or more software requirements, and of each software requirement to one or more CSC and CSCI tests

3. Configuration Management Principle

(No additional documents. See the Configuration Management KPA list.)

Quality Assurance KPA

1. Quality Assurance

a. Software Quality Assurance Plan

b. QA checklists

c. QA reports to levels above project management

d. QA nonconcurrence reports to project management

e. Policies and procedures for QA

f. Process and procedures for QA audits

Standards and Procedures KPA

1. **Standards and Procedures**
   a. Policies and procedures for policy/procedure definition management
   b. Specific policies and procedures for subcontract management, requirements management, QA, CM, and corrective action management. (Strengths if no software engineering process group (SEPG) or equivalent is identified.)
   c. SDP standards
   d. Software Quality Assurance Plan standards
   e. Software Configuration Management Plan standards
   f. Development standards (requirements, design, code, test, unit development folder)
   g. Procedure for enforcing compliance to standards

2. **Testing Approach**
   a. Test plan for a CSU, CSC, or CSCI, including functional test cases for all requirements, and white box test cases for boundary values, illegal values, worst-case scenarios, and branch testing
   b. Test plan peer review notes
   c. Test log
   d. Software development file
   e. Error reports

3. **Standards Principle**
   a. Software development standards (requirements, design, code, test, integration)
   b. Procedure for enforcing compliance to software standards
   c. Sample products covered by standards, such as requirements document, design document, source code listing, unit test plan, and integration test plan

4. **Documentation Principle**
   a. Standards for format of all deliverable software documentation

b. SDP - evaluation criteria for 1) correctness, completeness, consistency, and accuracy of delivered documents; 2) form and content of delivered documents; 3) sufficiency of documents as basis for subsequent life cycle activities

c. Software Requirements Document

d. Software Design Document

e. Source Code File

f. Software Test Plan

g. Software Test Procedures

h. Software Test Report

i. Software Transition Plan, to transition to support organization

j. Other support and operational documentation, such as installation and checkout procedures, configuration procedures, operation procedures, troubleshooting procedures, and shut-down procedures.

k. Software Development File (e.g., trade-off results, alternate designs, simulation and analysis results, prototyping results, peer review results, action items, problem reports, schedule and status information)

Peer Reviews KPA

1. **Peer Reviews**
    a. Procedures and practices for peer reviews
    b. Technical peer review schedule
    c. Peer review assignment list
    d. Peer review reports
    e. Peer review action items
    f. Statistics on conduct of peer reviews and on product errors detected
    g. Error detections checklists
2. **Shared Knowledge Principle**
    a. List of shared knowledge assignments

Training KPA

1. Training

a. Policies and procedures for training program, including training requirements, educational reimbursement, and professional society dues reimbursement

b. Course syllabus for developers, reviewers, managers, supervisors, quality assurance, configuration management

c. Course schedules

d. Organization training plan

e. Project training plan

f. Training records

Software Engineering Process Group KPA

1. **Software Engineering Process Group (SEPG)**
    a. SEPG charter
    b. Minutes of SEPG meetings
    c. Process assessment reports
    d. Process measurements
    e. SEPG plans for future improvements
2. **Secure Development Environment**
3. **Access Control Principle**
    a. Security policy
    b. Description of access control mechanisms (discretionary and mandatory), if available
4. **Auditing Principle**
    a. Computer security policy
    b. Audit trail of SEE activities
    c. Policy for storage of audit data
5. **Identification and Authentication Principle**
    a. Security policy
    b. Policy and procedure on authentication data
6. **Computer-Assisted Software Engineering (CASE) Tool Principle**
    a. Training schedule for CASE tools
    b. Training curriculum
    c. List of CASE tools in the SEE
    d. Sample products produced using CASE tools, such as requirements document, design document, traceability mapping, compilation listing, and test coverage analysis

7. Environment Administration Principle

a. Security policy

b. Environment Administration Procedures Document covering installation, configuration, operation, and maintenance procedures for all SEE components; including procedures for physical access to SEE facilities, SEE backup procedures, SEE recovery procedures after suspected subversion, disaster recovery plan, procedures for controlling modifications to SEE

c. Documentation on all known SEE flaws

d. SEE security analysis

8. Project Trust Policy

9. Trusted Distribution Principle

a. Documentation on trusted distribution mechanisms

b. Software Configuration Management Plan

10. Security Policy Principle

a. Security Policy and Procedures Document

b. Meeting notes from evaluations of security procedures

c. Security training or indoctrination materials for employees

11. Prototyping Principle

a. Plan for the use of each proof-of-concept prototype

b. SDP

12. Previously Developed Software Integrity Principle

a. SDP

b. Risk assessments and mitigations for previously developed software components

c. Document describing level of compliance to Trust Principles for each previously developed software component

LIST OF REFERENCES

[AFSC 1991] Air Force Systems Command. 1992. Pamphlet 800-5. *Acquisition Management: Software Development Capability Capacity Review.* Wright-Patterson AFB, OH.

[Bell 1992] Bell Canada. 1992. *Trillium: Telecom Software Product Development Capability Assessment Model.* Quebec, Canada.

[BMD 1993] Ballistic Missile Defense Organization. 1993. *Software Trust Principles.* Washington, DC.

[BMDO 1993] Ballistic Missile Defense Organization. 1993. Directive 3405. Washington, DC.

[Fife 1992] Fife, Dennis W. et. al. 1992. *Conducting Software Capability Evaluations.* IDA Paper P-2771. Alexandria, VA: Institute for Defense Analyses.

[GPALS 1992] Global Protection Against Limited Strikes. 1992. *GPALS Trusted Software Methodology, Volume 1.* SDI-S-SD-91-000007. Washington, DC.

[Humphrey 1987] Humphrey, W.S., and W.L. Sweet. 1987. *A Method for Assessing the Software Engineering Capability of Contractors.* SEI Technical Report SEI-87-TR-23. Pittsburgh, PA: Software Engineering Institute.

[Humphrey 1989] Humphrey, W.S. 1989. *Managing the Software Process.* Reading, MA: Addison-Wesley.

[ISO 1991] International Organization for Standardization. 1991. *ISO9000-3 Quality Management and Quality Assurance Standards Part 3: Guidelines for the Application of ISO 9001 to the Development, Supply and Maintenance of Software.* Gaithersburg, MD: National Institute of Standards and Technology.

[SEI 1992a] Software Engineering Institute. 1992. *Evaluation Team Training: Participant's Handbook*. Pittsburgh, PA.

[SEI 1992b] Software Engineering Institute. 1992. *Capability Maturity Model for Software*. Pittsburgh, PA.

[SPR 1991] Software Productivity Research. February 1991. *Checkpoint Questionnaire, Version 3.0*. Burlington, MA.

LIST OF ACRONYMS

BMD	Ballistic Missile Defense
BMDO	Ballistic Missile Defense Organization
CASE	Computer-Assisted Software Engineering
CCB	Configuration Control Board
CM	Configuration Management
CMM	Capability Maturity Model
CMU	Carnegie Mellon University
CSC	Computer Software Component
CSCI	Computer Software Configuration Item
CSU	Computer Software Unit
DoD	Department of Defense
FAR	Functional Area Representative
FFRDC	Federally Funded Research and Development Center
I/O	Input/Output
IDA	Institute for Defense Analyses
ISO	International Organization for Standardization
KPA	Key Process Area
MCCR	Mission Critical Computer Resources
N/A	Not Applicable
POC	Point of Contact
PM	Program Manager
PMM	Process Maturity Model
QA	Quality Assurance
RFP	Request for Proposal
S/W	Software
SCE	Software Capability Evaluation
SCM	Software Configuration Management
SDC/CR	Software Development Capability/Capacity Review
SDP	Software Development Plan

SEE	Software Engineering Environment
SEI	Software Engineering Institute
SEPG	Software Engineering Process Group
SPA	Software Process Assessment
SPR	Software Productivity Research
SQA	Software Quality Assurance
SSPM	Software Standards and Procedures Manual
SSEB	Source Selection Evaluation Board
TSDM	Trusted Software Development Methodology
TBD	To Be Determined
TSM	Trusted Software Methodology
VP	Vice-President

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	October 1994	Final

4. TITLE AND SUBTITLE
Software Capability Evaluations Incorporating Trusted Software Methodology

5. FUNDING NUMBERS
DASW01-94-C-0054

Task Order T-R2-597.2

6. AUTHOR(S)
Dennis W. Fife, Judy Popelas, Beth Springsteen

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Institute for Defense Analyses (IDA)
1801 N. Beauregard St.
Alexandria, VA 22311-1772

8. PERFORMING ORGANIZATION REPORT NUMBER
IDA Paper P-2895

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
BMDO/DB
Room 1E149, The Pentagon
Washington, D.C. 20301-7100

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, unlimited distribution: August 2, 1995.

12b. DISTRIBUTION CODE
2A

13. ABSTRACT (Maximum 200 words)

This paper defines a method for evaluating a contractor's ability to comply with both the Software Engineering Institute's (SEI) Process Maturity Model and the Ballistic Missile Defense's (BMD) Trusted Software Methodology (TSM). The SEI model is used to evaluate the maturity of a contractor's software development process. The TSM is used to evaluate the contractor's ability to prevent inadvertent and malicious errors from occurring in software products. The method described in this paper identifies the commonality and differences between these two models and the additional activities that must occur during an SEI Software Capability Evaluation (SCE) to accommodate the TSM criteria. This method was developed after receiving appropriate SCE and TSM training and performing numerous SCEs. It is intended to be used throughout the BMD program to help identify software development risks.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Software, Trusted Software, Evaluations, Models.	98
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	SAR